CLAIMS

1. A method for arbitrating between a plurality of access requests issued in relation to a resource by a plurality of requestors, wherein each request can be one of at least two types, a first of the types having a higher latency associated with its performance than at least some of the other types, the method including the steps of:

(a) receiving a plurality of the access requests; (the requests are not placed anywhere, they are simply received)

(b) maintaining a current pointer that points to a current timeslot in a timeslot list, and at least one lookahead pointer that points to a future timeslot in the timeslot list; and

(c) in the event an access request as arbitrated via the lookahead pointer is of the first type, initiating performance of the access request earlier than the position in the list suggests it would be performed should it be started when the current pointer reached the timeslot.

2. A method according to claim 1, wherein step (c) includes the substep of performing the access request indicated by the lookahead pointer immediately after the access request indicated by the current pointer is performed.

3. A method according to claim 1, wherein the first type of access request is a memory write request.

4. A method according to claim 1, wherein step (c) includes the substep of performing the access request indicated by the lookahead pointer immediately after the access request indicated by the current pointer is performed.

5. A method according to claim 1, wherein the number of timeslots between the timeslot indicated by the lookahead pointer and the timeslot indicated by the current pointer takes into account a latency difference between performing an access request of the first type and at least one of the other access request types.

6. An integrated circuit including:

a plurality of operative units, each of which is capable of issuing a request for access to a memory accessible by the integrated circuit; and

an timeslot arbitrator for arbitrating between requests issued by the operative units for access to the memory, wherein each request can be one of at least two types, a first of the types having a higher latency associated with its performance than at least some of the other types, the timeslot arbitrator being configured to:

(a) receive a plurality of the access requests;

(b) maintain a current pointer that points to a current timeslot in a timeslot list, and at least one lookahead pointer that points to a future timeslot in the timeslot list; and

(c) in the event the access request as arbitrated via the lookahead pointer is of the first type, performing the access request earlier than the position in the list suggests it should be performed should it be started when the current pointer reached the timeslot.

7. An integrated circuit according to claim 6, wherein the first type of access request is a memory write request.

8. An integrated circuit according to claim 7, wherein the integrated circuit includes a memory interface unit operatively connected with, and under the control of, the timeslot arbitrator, and wherein the memory interface is operatively connected to:

one or more of the operative units via one or more communications buses, and

the memory via a memory bus of greater width than the communications buses.

9. An integrated circuit according to claim 6, wherein the number of timeslots between the timeslot indicated by the lookahead pointer and the timeslot indicated by the current pointer takes into account a latency difference between performing an access request of the first type and at least one of access request types.

10. A method of arbitrating between access requests from a plurality of requestors for access to a resource, wherein at least one of the requestors is defined as higher priority access to the resource, the method comprising the steps of:

(a) receiving a plurality of the access requests;

(b) in the event an access request from the at least one of the requestors is received, initiating performance of the access request in preference to the requestor as specified by the timeslot list and regardless of whether or not the at least one of the requestors is in the timeslot list.

11. A method according to claim 10, wherein the at least one requestor requires lower latency access to the resource than at least one of the other requestors from which access requests can be received.

12. A method according to claim 10, wherein the at least one requestor is a processor.

13. A method according to claim 10, wherein the resource is a memory.

14. A method according to claim 10, wherein step (b) includes the substep of performing the access request from the requestor immediately following completion of any current access request being reformed.

15. A method according to claim 10, wherein step (b) is performed such that a frequency of the at least one requestor being granted preferential performance of its access requests is limited within a time period.

16. A method according to claim 15, wherein early performance of access requests from the at least one requestor is restricted to a maximum number of times within a predetermined number of timeslots.

17. A method according to claim 10, wherein the requestors are hardware units on an integrated circuit and the method is implemented by a timeslot arbitrator unit on the integrated circuit.

18. A method according to claim 10, wherein each request can also be one of at least two types, a first of the types having a higher latency associated with its performance than at least some of the other types, the method including the steps of:

(a) receiving a plurality of the access requests;

(b) maintaining a current pointer that points to a current timeslot in the timeslot list, and at least one lookahead pointer that points to a future timeslot in the timeslot list; and

(c) in the event an access request as arbitrated via the lookahead pointer is of the first type, initiating performance of the access request earlier than its position in the list suggests it should be performed should it be started when the current pointer reached the timeslot.